(NASA-CR-139375) ON-LINE DIAGNOSIS OF UNRESTRICTED FAULTS (Michigan Univ.) 35 p HC $4.75 CSCL 09B

N74-29529

Unclas

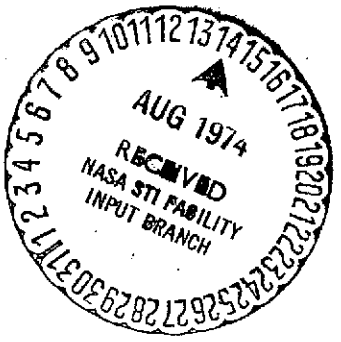
G3/08 44752

On-Line Diagnosis of Unrestricted Faults

John F. Meyer
Senior Member, IEEE

and

Robert J. Sundstrom
Member, IEEE



Index Terms

On-line diagnosis, concurrent error detection, fault diagnosis, reliable automata, unrestricted faults, inverse sequential machines.

Abstract

A formal model for the study of on-line diagnosis is introduced and used to investigate the diagnosis of unrestricted faults. Within this model a fault of a system S is considered to be a transformation of S into another system S' at some time τ. The resulting faulty system is taken to be the system which looks like S up to time τ and like S' thereafter. Notions of fault tolerance and error are defined in terms of the resulting system being able to mimic some desired behavior as specified by a system $\tilde{S}$. A notion of on-line diagnosis is formulated which involves an external detector and a maximum time delay within which every error caused by a fault in a prescribed set must be detected.

The set of unrestricted faults of a system is defined to be simply the set of all faults of that system. It is shown that if a system is on-line diagnosable for the unrestricted set of faults then the detector is at least as complex, in terms of state set size, as the specification. Moreover, this is true even if an arbitrarily large delay is allowed in the diagnosis. The use of inverse systems for the diagnosis of unrestricted faults is considered. A partial characterization of those inverses which can be used for unrestricted fault diagnosis is obtained.

I. INTRODUCTION

In many applications, especially those in which a computer is being used to control some process in real-time, (e.g., telephone switching, flight control of an aircraft or spacecraft, etc.) it is desirable to constantly monitor the performance of the system, as it is being used, to determine whether the actual system is within tolerance of the intended system. Informally, by "on-line diagnosis" we mean a monitoring process of this type where the extent of the diagnosis depends on the meaning of "within tolerance." Thus, for example, if being within tolerance means having the same input-output behavior, then on-line diagnosis becomes on-line "detection." In the special case where the implementation of on-line diagnosis is completely internal to the system being diagnosed, it is referred to as "self diagnosis" or "self checking."

The incorporation of special hardware for the purpose of on-line diagnosis dates way back to the relay computers developed by Bell Laboratories in the early-to-mid 1940's, where biquinary codes were used to dynamically check the operation of the computer [1]. A more general look at codes for checking logical operations was first taken by Peterson and Rabin in 1959 [2] where they showed that combinational circuits can vary greatly in their inherent on-line diagnosability. The use of coding techniques in the design of self-checking circuits was further explored by Carter and Schneider in 1968 [3] and by Anderson in 1971 [4]. In addition, a number of special on-line diagnosis methods have been considered which apply to specific hardware subsystems such as adders, counters, etc. (see [5], for example).

Given this background of techniques that have been proposed and, in many cases, used to improve the on-line diagnosability of a system, the following question arises quite naturally. With regard to <u>any</u> technique that might be employed, how complex must the diagnosing system be as compared to the system being

diagnosed, if the latter is to be on-line diagnosable in some prescribed sense? To answer this question, one must, of course, designate the class of systems considered, the complexity measure, and the precise meaning of on-line diagnosis. In a first attempt, it appears reasonable to make these devices as general as possible in order to establish a framework for more incisive results that might follow.

Specifically, the systems we have chosen to consider are those which are representable as "discrete-time" systems when subjected to transient or permanent faults. Such systems are generalizations of sequential machines and permit structure to vary as faults occur. As a measure of system complexity, we have chosen the number of reachable internal states. This measure reflects the memory capacity of a system and, without further restrictions on system structure, it's the only measure of structural complexity that has a reasonable interpretation. Finally, the concept of on-line diagnosis considered requires that any error caused by a fault be detected within some maximum allowable time delay.

Section II of the paper is concerned with the formal development of the notion of a discrete-time system and the associated concepts of fault, result of a fault, and error. Section III formalizes the above concept of on-line diagnosis and establishes an answer to the question posed above; namely, if no restrictions are placed on the potential faults of a system S, then the complexity of a detector D must be at least as great as that of S. Moreover, this result holds even when the allowed time delay for error detection is arbitrarily large. Section IV considers the on-line diagnosis of unrestricted faults for systems which have (delayed) inverses, that is, systems which are information lossless. Here it is shown that an inverse system can always be used for on-line diagnosis if it too is information lossless. Although the lossless condition is sufficient, it is shown further that there exist systems for which a lossy inverse can also be used for on-line diagnosis.

II. FAULTS AND ERRORS IN DISCRETE-TIME SYSTEMS

Informally, a discrete-time system is a causal, deterministic, finite-state system to which inputs (from a finite set) are applied at discrete instants of time and from which states and outputs (from a finite set) are observed at discrete instants of time. If, in addition, specific inputs are designated as "reset" inputs (used to initialize the system), then discrete time systems can be formally defined as follows.

Definition 1: Relative to the time-base $T = \{\ldots -1,0,1,\ldots\}$, a (resettable) discrete-time system (with finite input, output, and reset alphabets) is a system

$$S = (I,Q,Z,\delta,\lambda,R,\rho)$$

where

I is a finite nonempty set, the input alphabet
Q is a finite nonempty set, the state set
Z is a finite nonempty set, the output alphabet
δ: $Q \times I \times T \longrightarrow Q$, the transition function
λ: $Q \times I \times T \longrightarrow Z$, the output function
R is a finite nonempty set, the reset alphabet
ρ: $R \times T \longrightarrow Q$, the reset function.

The first five elements, I, Q, Z, δ, and λ, of a discrete-time system are the usual elements of a sequential machine but with δ and λ generalized to account for possible variation of structure with time. The action of a reset $r \in R$ is described by ρ, the reset function, with the interpretation that if reset r is applied at time $t - 1$ then the system will be in state $\rho(r,t)$ at time t. In the special case where S is time-invariant we will adopt the usual terminology by referring to S as a (resettable) sequential machine.

A particular discrete-time system can be viewed as a system which looks like some sequential machine S_1 in one time interval, like S_2 in another interval, and so on (see Fig. 1). Assuming familiarity with the concept of a sequential machine, with this view the more general concept of discrete-time system is easily understood. Moreover, as will be observed in the discussion that follows, discrete-time systems suffice to represent the structure and behavior of both "fault-free" and "faulty" digital systems in an on-line diagnosis environment.

Formulation of an appropriate notion of behavior for discrete-time systems follows directly from the usual behavioral notions that have been considered for sequential machines. Informally, if S is a discrete-time system, the behavior of S for a reset r applied at time t is a function which maps an input sequence x into the last output symbol that S would emit given that it received x under the above conditions. More formally, the behavior of S for (initial) condition (r,t) ($r \in R$, $t \in T$) is the function

$$\beta_{r,t}: \quad I^+ \longrightarrow Z$$

where

$$\beta_{r,t}(x) = \bar{\lambda}(\rho(r,t), x, t) \tag{2.1}$$

($\bar{\lambda}$ denotes the natural extension of λ to $Q \times I^+ \times T$.) The natural extension of $\beta_{r,t}$ to sequences is denoted by $\hat{\beta}_{r,t}$, that is,

$$\hat{\beta}_{r,t}: \quad I^+ \longrightarrow Z^+$$

where

$$\hat{\beta}_{r,t}(a_1 a_2 \cdots a_n) = \beta_{r,t}(a_1)\beta_{r,t}(a_1 a_2)\cdots\beta_{r,t}(a_1 a_2 \cdots a_n).$$

It will also be convenient to define the behavior of S in state q, that is, the function

$$\beta_q: \quad I^+ \times T \longrightarrow Z$$

where

$$\beta_q(x,t) \quad = \quad \overline{\lambda}(q,x,t).$$

Given a discrete-time system S, the reachable part of S is the set

$$P \quad = \quad \{q \in Q \mid q \quad = \delta(\rho(r,t),x,t) \text{ for some } r \in R, t \in T, \text{ and } x \in I^*\}.$$

($\overline{\delta}$ denotes the natural extension of δ to $Q \times I^* \times T$.) S is reachable if $P = Q$. S is reduced if for all $q,q' \in P$, $\beta_q = \beta_{q'}$ implies $q = q'$. Concepts of simulation and realization that have been considered for sequential machines (see [6], for example) also extend easily to discrete-time systems. In particular, given two systems S and $\tilde{S}$, S realizes $\tilde{S}$ under (g,h,k) if $g: \quad (\tilde{I})^+ \longrightarrow I^+$ is a semigroup homomorphism such that $g(\tilde{I}) \subseteq I$, $h: \quad \tilde{R} \longrightarrow R$, and $k: \quad Z' \longrightarrow \tilde{Z}$ where $Z' \subseteq Z$ such that for all $\tilde{r} \in \tilde{R}$, and $t \in T$

$$\tilde{\beta}_{\tilde{r},t} \quad = \quad k \circ \beta_{h(\tilde{r}),t} \circ g \qquad (2.2)$$

(where $\circ$ denotes left composition of functions). A pictorial representation of this notion is given in Fig. 2. A realization concept is quite useful when considering questions of diagnosability, for one often begins with a system specification $\tilde{S}$ which describes what the user wants but is not diagnosable. The solution is to find another system S which is diagnosable and can realize the behavior of $\tilde{S}$ via the input encoding map g, the reset encoding map h, and the output decoding map k.

Given some discrete-time system S, let us now consider how faults effect changes in system structure. In general, if a fault occurs at some time τ, S will be transformed into some other system S' and if S is in state q just before τ then S' is in state q' just after τ. More formally, a fault of S is a triple $f = (S', \tau, \theta)$ where S' has the same input, output, and reset alphabets as S, $\tau \in T$, and θ: $Q \longrightarrow Q'$. The restriction on the input, output, and reset alphabets is reasonable since after the fault occurs the system will presumably have the same external terminals. The function θ describes the state transitions that result when the fault occurs. Note that the interpretation of fault here is one of effect, not cause. Thus, for example, if S represents a switching network and some gate output j becomes stuck-at-1 at time τ, the fault is represented by the triple $f = (S', \tau, \theta)$ where S' represents the network, as modified by a constant 1 at output j, and θ describes how this change affects the next state.

Given this interpretation, a formulation of the resulting faulty system is straightforward. More precisely,

Definition 2: If $f = (S', \tau, \theta)$ is a fault of S, the result of f is the system

$$S^f = (I, Q^f, Z, \delta^f, \lambda^f, R, \rho^f)$$

where

$$Q^f = Q \cup Q'$$

$$\delta^f(q,a,t) = \begin{cases} \delta(q,a,t) & \text{if } q \in Q \text{ and } t < \tau - 1 \\ \theta(\delta(q,a,t)) & \text{if } q \in Q \text{ and } t = \tau - 1 \\ \delta'(q,a,t) & \text{if } q \in Q' \text{ and } t \geq \tau \end{cases}$$

$$\lambda^f(q,a,t) = \begin{cases} \lambda(q,a,t) & \text{if } q \in Q \text{ and } t < \tau \\ \lambda'(q,a,t) & \text{if } q \in Q' \text{ and } t \geq \tau \end{cases}$$

$$\rho^f(r,t) = \begin{cases} \rho(r,t) & \text{if } t < \tau \\ \theta(\rho(r,t)) & \text{if } t = \tau \\ \rho'(r,t) & \text{if } t > \tau \end{cases}$$

(Arguments not specified in the above definitions may be assigned arbitrary values.) A pictorial view of the result of f is presented in Fig. 3.

Given the result S^f of some fault f, the behavior of S^f for initial condition (r,t) (see (2.1)) can be conveniently formulated as follows.

Theorem 1: Let S be a system and let $f = (S',\tau,\theta)$ be a fault of S. Then for each $r \in R$, $t \in T$, and $x \in I^+$

$$\beta^f_{r,t}(x) = \begin{cases} \beta_{r,t}(x) & \text{if } t + |x| \leq \tau \\ \beta'_{\theta(\bar{\delta}(\rho(r,t),y,t))}(z,\tau) & \text{if } t+|x| > \tau \text{ and } t \leq \tau \text{ where } x=yz \text{ and } |y| = \tau - t \\ \beta'_{r,t}(x) & \text{if } t > \tau \end{cases}$$

($|x|$ denotes the length of sequence x.)

The proof of Theorem 1 is a straightforward application of the general definition of behavior (2.1) to the faulty system S^f given by Definition 2. Its utility is that it provides a formal means for comparing the behavior of a faulty system S^f to that of the fault-free system S or to that of some original specification $\tilde{S}$. In particular, we want to determine whether the behavior of S^f is "within tolerance" of the specification $\tilde{S}$. The latter concept can be formalized as follows.

Let $\tilde{S}$ be a reduced, reachable specification of a time-invariant, discrete-time system (i.e., $\tilde{S}$ is a sequential machine) and let S be a sequential machine that realizes $\tilde{S}$ under the functions (g,h,k). (Our development at this point could be generalized to include time-varying systems. However, it seems reasonable to assume that the specification and desired fault-free realization are time-invariant.) We can assume further that g and h are onto since the only input and reset symbols of concern in the realization S are those which correspond to inputs and resets of $\tilde{S}$. Also, since $\tilde{S}$ and S are time-invariant, it suffices to describe their behaviors for resets at time 0. Accordingly, we will let $\tilde{\beta}_r$ and β_r denote the behaviors $\tilde{\beta}_{r,0}$ and $\beta_{r,1}$ respectively.

Given the above assumptions, we will say that a faulty system S^f is "within tolerance" of S or alternatively, that the fault f is "tolerated" if, behaviorally, S^f relates to $\tilde{S}$ in the same way that S relates to $\tilde{S}$. In other words, behaviorally, S and S^f can accomplish the same thing relative to $\tilde{S}$. (Note that although S is presumed time-invariant, in general, S^f will not be.) More formally, if f is a fault of machine S, then f is <u>tolerated</u> if, for all $\tilde{r} \in \tilde{R}$,

$$\tilde{\beta}_{\tilde{r}} = k \circ \beta^f_{h(\tilde{r})} \circ g \ .$$

Alternatively, since g and h are onto, it follows that f is tolerated if and only if, for all $r \in R$,

$$k \circ \beta_r = k \circ \beta^f_r \ .$$

A fault which is not tolerated is capable of causing "errors" in the following sense. If $r \in R$, $x \in I^+$ and $y \in Z^+$

such that $|x| = |y|$, the triple (r,x,y) is an error if

$$\bar{k}(\beta_r(x)) \neq \bar{k}(y)$$

where $\bar{k}$ denotes the homomorphic extension of k to Z^+. In particular, if f is a fault, an error (r,x,y) is caused by f if

$$\hat{\beta}_r^f(x) = y$$

that is, for reset r and input sequence x, S^f produces an output that is in error relative to $\tilde{S}$. It follows immediately from the definition that a fault f is tolerated if and only if no errors are caused by f. Finally, since we will be interested in the time when an error first occurs, we will say that an error (r,ua,vb) (where $r \in R$; $u, v \in I^+$; $a, b \in I$) is minimal if (r,u,v) is not an error.

III. ON-LINE DIAGNOSIS

With respect to the concepts of fault and error developed in the preceding section, let us now consider what we might mean by "on-line diagnosis." Let (S,F) be the machine S along with the prescribed set of faults F of S. Let D be another machine with the same reset alphabet as S and with input set $Z \times I$ and let n be a nonnegative integer. Then

Definition 3: (S,F) is (D,n)-diagnosable if

(i) $\beta_r^D([\hat{\beta}_r(x), x]) = 0$ for all $r \in R$ and $x \in I^+$ and

(ii) if (r,x,y) is a minimal error caused by some $f \in F$ then

$$\hat{\beta}_r^D([\hat{\beta}_r^f(xw), xw]) \neq 0^{|xw|} \text{ for all } w \in I^* \text{ with } |w| = n.$$

(If $u = z_1 z_2 \ldots z_n \in Z^+$ and $v = a_1 a_2 \ldots a_n \in I^+$ then $[u,v]$ denotes the sequence $(z_1,a_1)(z_2,a_2)\ldots(z_n,a_n) \in (Z \times I)^+$.)

Thus, the detector D observes the operation of S^f (see Fig. 4) and must make a decision, based on this observation, as to whether an error has occurred. Note that the fault-free realization S and the detector are both time-invariant (i.e., machines), and that the detector takes no part in the computation of S's output. The two conditions of the above definition can be paraphrased as:

(i) D responds negatively if no fault occurs, i.e., D gives no false alarms; and

(ii) for all $f \in F$, D responds positively within n time steps of the occurrence of the first error caused by f.

Given this concept of on-line diagnosability, the investigation that follows will be concerned with the general case in which the set of potential faults is "unrestricted." More precisely, the set of unrestricted faults of machine S, denoted by U, is the set $U = \{f | f \text{ is a fault of } S\}$. Note that this set of faults is truly unrestricted for it is precisely the set of all possible faults of the machine being diagnosed.

Aside from representing a "worst-case" fault environment, there are certain practical reasons for considering U, at least at the outset. In particular, as the scale of integrated circuit technology becomes larger, it becomes more difficult to postulate a suitably restricted class of faults such as the class of all

"stuck-at" faults. Moreover, although other failure models such as bridging failures have been proposed and studied (see [7] and [8] for example), little is known about the diagnosis of such failures. In addition, intermittent and multiple failures are also possible and are even more difficult to model. Finally, for a given failure it may be impossible to determine the θ function of the fault caused by this failure. Thus fault sets which do not restrict the fault mapping θ are advantageous.

One important property of the set of unrestricted faults is the relation between this fault set and the set of errors that may be caused by faults in this set. Given any $r \in R$, $x \in I^+$, and $y \in Z^+$ with $|x| = |y|$, there is a fault $f \in U$ such that $\hat{\beta}_r^f(x) = y$. Therefore faults in U can cause any possible erroneous behavior, and for (S,U) to be (D,n)-diagnosable all of these possible erroneous behaviors will have to be detected by D. Due to the above observation it is clear that the output of S^f (the system actually being observed by the detector) can give no information about what the correct output should be.

It is a well known and obvious fact that if a system is duplicated and both copies are run in parallel with the same inputs, then, by dynamically comparing the outputs on the two copies, any error which does not appear simultaneously in both copies will be immediately detected. Our view of duplication is shown in Fig. 5. In this figure the detector D consists of a copy of S along with a generalized Exclusive-OR gate whose output is 0 if and only if its inputs are identical. Given such a detector D, it is immediately clear that (S,U) is (D,0)-diagnosable. It is also clear that by using suitable encoding and decoding functions, unrestricted fault diagnosis can be achieved by comparing the output of S with that of its reduced and reachable specification $\tilde{S}$.

An interesting question, the answer to which would tell us something fundamental about the diagnosis of unrestricted faults, is whether or not it is possible to do better than duplication in the sense of achieving (D,n)-diagnosis of (S,U) with a detector D which is less complex, in terms of state set size, than the specification $\tilde{S}$. One reason to believe that this may be possible is the observation that if $\tilde{S}$ has an inverse then this inverse may have fewer states than $\tilde{S}$ and yet a detector constructed using this inverse may be capable of diagnosing the set of unrestricted faults of S (see Example 1). However, the following result shows that for n = 0 it is impossible to do any better than duplication in the sense described above. First we state a simple lemma which is an immediate consequence of the definition of realization (2.2).

Lemma 1: Let S and $\tilde{S}$ be two machines such that S realizes $\tilde{S}$ under the triple (g,h,k) and $\tilde{S}$ is reduced and reachable. Then there exists a 1-1 function σ from $\tilde{Q}$ into P such that for all $q \in \tilde{Q}$, $\tilde{\beta} = k \circ \beta_{\sigma(q)} \circ g$.

Applying this lemma, we obtain the following basic result.

Theorem 2: If (S,U) is (D,0)-diagnosable, then the detector D has at least as many states as the specification $\tilde{S}$ of S.

Proof: Let (S,U) be (D,0)-diagnosable and assume, to the contrary, that $|Q_D| < |\tilde{Q}|$. By the above lemma, there are $|\tilde{Q}|$ states in P, the reachable part of S, which all mimic different states of $\tilde{S}$. Referring to Fig. 4, since $|Q_D| < |\tilde{Q}|$ there must exist states q_1, $q_2 \in P$ and $s \in Q_D$ such that $k \circ \beta_{q_1} \neq k \circ \beta_{q_2}$, and it is possible for S to be in q_1 or q_2 while D is in s. Since $k \circ \beta_{q_1} \neq k \circ \beta_{q_2}$, there exists a sequence ua where $u \in I^*$ and $a \in I$ such that $k(\beta_{q_1}(ua)) \neq k(\beta_{q_2}(ua))$ and if $u \neq \Lambda$ then $\bar{k}(\hat{\beta}_{q_1}(u)) = \bar{k}(\hat{\beta}_{q_2}(u))$. Since it is possible for S to be in q_1 while D is in s, there exists $r_1 \in R$ and $x_1 \in I^+$ such that $\delta(\rho(r_1), x_1) = q$ and $\delta_D(\rho_D(r_1), [\hat{\beta}_r(x_1), x_1]) = s$.

Recall that given any $r \in R$, $x \in I^+$, and $y \in Z^+$ with $|x| = |y|$, there is a fault $f \in U$ such that $\hat{\beta}_r^f(x) = y$. Let $f \in U$ be a fault for which $\hat{\beta}_{r_1}^f(x_1ua) = \hat{\beta}_{r_1}(x_1)\hat{\beta}_{q_2}(ua)$. Since it is known that $\bar{k}(\hat{\beta}_{q_1}(u)) = \bar{k}(\hat{\beta}_{q_2}(u))$, it follows that $(r_1, x_1ua, \hat{\beta}_r^f(x_1ua))$ is a minimal error. Now (S,U) is (D,0)-diagnosable implies $\hat{\beta}_{r_1}^D([\hat{\beta}_{r_1}^f(x_1ua), x_1ua]) \neq 0^{|x_1ua|}$. Since no false alarms may occur, $\hat{\beta}_{r_1}^D([\hat{\beta}_{r_1}(x_1), x_1]) = 0^{|x_1|}$. Also, since it is possible for S to be in q_2 while D is in s, $\hat{\beta}_s^D([\hat{\beta}_{q_2}(ua), ua]) = 0^{|ua|}$. But

$$\hat{\beta}_{r_1}^D([\hat{\beta}_{r_1}^f(x_1ua), x_1ua]) = \hat{\beta}_{r_1}^D([\hat{\beta}_{r_1}(x_1)\hat{\beta}_{q_2}(ua), x_1ua])$$

$$= \hat{\beta}_{r_1}^D([\hat{\beta}_{r_1}(x_1), x_1])\hat{\beta}_s^D([\hat{\beta}_{q_2}(ua), ua])$$

$$= 0^{|x_1|}0^{|ua|}$$

$$= 0^{|x_1ua|}$$

This contradicts the assumption that (S,U) is (D,n)-diagnosable. Therefore $|Q_D| \geq |\tilde{Q}|$, thus completing the proof.

Suppose now that we allow some arbitrary, but fixed, $n > 0$ in the detection process. Can this additional time be traded off for less detector complexity? Unfortunately, for the unrestricted case, the answer is no. In fact, if (S,U) is (D',n)-diagnosable then we can construct a detector D, essentially by eliminating unnecessary states of D', such that (S,U) is (D,0)-diagnosable.

Before stating this result formally, it is convenient to establish the following important lemma.

Lemma 2: If (S,U) is (D',n)-diagnosable then there exists a detector D with no more states than D' such that (S,U) is (D,n)-diagnosable and, for each $q \in Q_D$, $\lambda_D(Q, (z,a)) = 0$ for some $(z,a) \in Z \times I$.

Proof: Assume that (S,U) is (D',n)-diagnosable and construct D from D' as follows:

1) Delete from the state table of D' any row corresponding to a state q for which

$$0 \notin \{\lambda_{D'}(q,(z,a)) \mid (z,a) \in Z \times I\} .$$

2) In the resulting table, replace every reference to the deleted state with a reference to an arbitrary remaining state, and set the corresponding output to 1.

3) Repeat steps 1) and 2) until no further deletions are possible.

Since $|Q_{D'}| < \infty$ the above algorithm will terminate in a finite number of iterations.

From the nature of the above construction it is clear that $|Q_D| \leq |Q_{D'}|$ and for each $q \in Q_D$, $\lambda_D(q, (z,a)) = 0$ for some $(z,a) \in Z \times I$. It only remains to be shown that (S,U) is (D,n)-diagnosable.

If the detector D' is in a state q for which $0 \notin \{\lambda_{D'}(q, (z,a)) \mid (z,a) \in Z \times I\}$, then an error must have occurred because if D' is in q then an error detection signal will be emitted regardless of the input to D'. Hence this error could be signaled whenever a transition to q is indicated, and there would be no loss in diagnosis and no possibility for a false alarm. Since all minimal errors which q signaled would then be signaled before D' got to state q', q' could be eliminated. This is the essence of what is accomplished in

steps 1) and 2). This elimination process is necessarily iterative because step 2) may introduce new states to be deleted. Since this construction is diagnosis preserving, (S,U) is (D,n)-diagnosable, thereby proving the lemma.

Theorem 3: If (S,U) is (D',n)-diagnosable then there exists a detector D with no more states than D' such that (S,U) is (D,0)-diagnosable.

Proof: Assume that (S,U) is (D',n)-diagnosable. By Lemma 2, there exists a detector D with no more states than D' such that (S,U) is (D,n)-diagnosable and, for each $q \in Q_D$, $\lambda_D(q, (z,a)) = 0$, for some $(z,a) \in Z \times I$.

Claim: (S,U) is (D,0)-diagnosable.

Assume, to the contrary, that (S,U) is not (D,0)-diagnosable. Using induction on the delay of the diagnosis, we will deduce that (S,U) is not (D,m)-diagnosable for all $m \geq 0$. This will establish the result for it contradicts the hypothesis that (S,U) is (D,n)-diagnosable.

If $m = 0$, then by the above assumption, (S,U) is not (D,m) diagnosable. Let us assume, then, that (S,U) is not (D,m)-diagnosable for some $m \geq 0$, and show that this implies (S,U) is (D,m+1)-diagnosable. Since (S,U) is not (D,m)-diagnosable, there exists a minimal error (r,x,y) caused by $f \in U$ and a sequence $w \in I^+$ with $|w| = m$ such that $\hat{\beta}_r^D([\hat{\beta}_r^f(xw), xw]) = 0^{|xw|}$. Let $\delta_D(\rho_D(r), [\hat{\beta}_r^f(xw), xw]) = s$. Let $(z,a) \in Z \times I$ such that $\lambda_D(s,(z,a)) = 0$. By Lemma 2 we know that such a (z,a) exists. Let f' be a fault for which $\hat{\beta}_r^{f'}(xwa) = \hat{\beta}_r^f(xw)z$. Then $(r,x,\hat{\beta}_r^{f'}(x))$ is a minimal error but $\hat{\beta}_r^D([\hat{\beta}_r^{f'}(xwa), xwa]) = 0^{|xwa|}$. Hence (S,U) is not (D,m+1)-diagnosable. Therefore, (S,U) is not (D,0)-diagnosable implies (S,U) is not (D,m)-diagnosable for all $m \geq 0$.

But we know that (S,U) is (D,n)-diagnosable. Hence (S,U) is (D,0)-diagnosable. This establishes the result.

Corollary 3.1: If (S,U) is (D,n)-diagnosable then the detector D has at least as many states as the specification $\tilde{S}$ of S.

Proof: This is an immediate consequence of Theorems 2 and 3.

IV. DIAGNOSIS USING INVERSE MACHINES

Let us now consider the use of inverse machines for the diagnosis of unrestricted faults. An (I,n)-delay machine (delay machine) is a machine $S^n = (I, I^n, I, \delta, \lambda, R, \rho)$ such that if $a_i \in I$, $1 \leq i \leq n+1$, then

$$\delta((a_1,\ldots,a_n), a_{n+1}) = (a_2,\ldots,a_{n+1})$$

and $$\lambda((a_1,\ldots,a_n), a_{n+1}) = a_1$$

thus, an (I,n)-delay machine simply deplays its input for n time steps. Stated more precisely, if S^n is an (I,n)-delay machine then

$$\beta^n_{(a_1,\ldots,a_n)}(a_{n+1},\ldots,a_{n+m}) = a_m$$

Let S and $\bar{S}$ be two machines such that $R = \bar{R}$ and $Z = \bar{I}$ Then $\bar{S}$ is an (n-delayed) inverse of S if there exists an (I,n)-delay machine S^n with reset alphabet R such that for all $r \in R$ and $x \in I^+$

$$\bar{\beta}_r(\hat{\beta}_r(x)) = \beta_r^n(x).$$

Machines for which inverses exist can be easily characterized. Intuitively, such machines lose no information as they transform input sequences into output sequences. A machine S is information lossless of delay d if for all $r \in R$ and $a_1a_2\ldots a_n$, $b_1b_2\ldots b_n \in I^+$ ($a_i, b_i \in I$, $1 \leq i \leq n$)

$$\hat{\beta}_r(a_1a_2\ldots a_n) = \hat{\beta}_r(b_1b_2\ldots b_n) \text{ implies } a_i = b_i$$

for $1 \leq i \leq n-d$.

The basic relationship between information losslessness and inverses is given by the following theorem (see [10], for example).

Theorem 4: S has an n-delayed inverse if and only if S is information lossless of delay n.

Information lossless machines and inverse machines were first introduced by Huffman [9]. He devised a test for information losslessness and for the existence of inverses. It should be pointed out that our definition of these notions are oriented towards their use in diagnosis and that they vary slightly from Huffman's definitions.

Even [10] later devised a better means of determining information losslessness and he presented two means for obtaining inverses of information lossless machines. Kohavi and Lavallee [11] have shown that any machine can be realized by an information lossless machine.

We now state the basic result relating the use of lossless inverses with the diagnosis of unrestricted faults.

Theorem 5: Let S be a lossless machine and let $\bar{S}$ be an n-delayed inverse of S. Let D be constructed from $\bar{S}$, the

(I,n)-delay machine which demonstrates that $\overline{S}$ is an n-delayed inverse of S, and an Exclusive-OR gate as shown in Fig. 6. If $\overline{S}$ is lossless of delay d then (S,U) is (D,d)-diagnosable.

Proof: Since $\overline{\beta}_r(\hat{\beta}_r(x)) = \beta_r^n(x)$, there will be no false alarms.

Let (r,x,y) be a minimal error caused by a fault $f \in U$. Then $\beta_r^f(x) = \beta_r(x)$. Let $w \in I^*$ with $|w| = d$. Since $\overline{S}$ is lossless of delay d, $\hat{\overline{\beta}}_r(\hat{\beta}_r^f(xw)) \neq \hat{\overline{\beta}}_r(\hat{\beta}_r(xw))$. The Exclusive-OR gate will detect this inequality, and hence the minimal error will be detected within d time steps of its occurrence. Therefore, (S,U) is (D,d)-diagnosable.

Example 1: Consider the reduced and reachable machines S_1 and $\overline{S}_1$ given by the state tables in Fig. 7 and Fig. 8. The last column in these state tables specifies the reset alphabet and function. $\overline{S}_1$ is a 2-delayed inverse of S_1 and $\overline{S}_1$ is itself information lossless of delay 2. Thus a detector D_1 for which (S_1,U) is $(D_1,2)$-diagnosable can be constructed using the inverse $\overline{S}_1$ of S_1.

It is interesting to note that although $\overline{S}_1$ has fewer states than S_1, D_1 has more states than S_1. This is because there is an $(I_1,2)$-delay machine in D_1, in addition to the inverse $\overline{S}_1$. It is also worth pointing out that the delay in diagnosis using an inverse machine is not the delay of losslessness of the machine being diagnosed but rather of its inverse. Thus an n-delayed inverse can be used to achieve diagnosis without delay if it is lossless of delay 0.

The following example shows that the converse of Theorem 5 does not hold. Namely, it is possible to diagnose the unrestricted faults of a machine using an inverse which is not lossless. However, not all inverses can be used for the diagnosis of unrestricted faults. The complete characterization of inverses which can be used for unrestricted fault diagnosis is still an open problem.

Example 2: Consider the reduced and reachable machines S_2 and $\overline{S}_2$ given by the state tables in Fig. 8 and Fig. 9. $\overline{S}_2$ is a 0-delayed inverse of S_2 and it can be used to construct a detector D_2 such that (S_2,U) is $(D_2,0)$-diagnosable. However, $\overline{S}_2$ is not lossless.

In conclusion, it is interesting to note that results established in this and the preceding section have something to say about lossless machines, per se. Let S be reduced, reachable, and lossless of delay d machine. Let $\overline{S}$ be a lossless inverse of S. We have seen in Example 1 that such an inverse can have fewer states than the machine of which it is an inverse. In the following result we will give a lower bound on the state set size of $\overline{S}$ in terms of state set size of S, the delay d of S, and the input alphabet size of S. This result, which deals only with lossless and inverse machines is proved using Corollary 3.1 and Theorem 5, results concerning the diagnosis of unrestricted faults.

Theorem 6: Let S be reduced, reachable, and lossless of delay d. Let $\overline{S}$ be a lossless d-delayed inverse of S. Then

$$|\overline{Q}| \geq \frac{|Q|}{|I|^d}$$

Proof: Consider S and $\overline{S}$ in the configuration of Fig. 6. Since $\overline{S}$ is lossless, by Theorem 5, (S,U) is (D,n)-diagnosable for some n. Now by Corollary 3.1 $|Q| \leq |Q_D|$. Since

$$Q_D = \overline{Q} \times I^d, \quad |Q_D| = |\overline{Q}||I|^d$$

Thus

$$|Q| \leq |\overline{Q}||I|^d \text{ or } \frac{|Q|}{|I|^d} \leq |\overline{Q}| \quad .$$

J. F. Meyer is with the Department of Computer and Communication Sciences and the Department of Electrical and Computer Engineering (Program in Computer, Information, and Control Engineering), The University of Michigan, Ann Arbor, Michigan.

R. J. Sundstrom is with the Program in Computer, Information, and Control Engineering, The University of Michigan, Ann Arbor, Michigan.

This work was supported in part by the National Aeronautics and Space Administration under grant NGR-23-005-622.

REFERENCES

[1] S. B. Williams, "Bell Telephone Laboratories Relay Computing System," Ann. Computation Lab. Harvard Univ., vol. XVI, pp. 41-53, 1948.

[2] W. W. Peterson and M. O. Rabin, "On codes for checking logical operations," IBM Journal, vol. 3, pp. 163-168, April 1959.

[3] W. C. Carter and P. R. Schneider, "Design of dynamically checked computers," Proc. of the IFIPS, Edinburgh, Scotland, pp. 878-883, Aug. 1968.

[4] D. A. Anderson, "Design of self-checking digital networks using coding techniques," Coordinated Sci. Lab., Univ. of Illinois, Urbana, Report R-527, Sept. 1971.

[5] F. F. Sellers, M. Hsiao, and L. W. Bearnson, Error Detection Logic for Digital Computers, McGraw-Hill, New York, 1968.

[6] J. F. Meyer and B. P. Zeigler, "On the limits of linearity," Theory of Machines and Computation (Edited by Z. Kohavi and A. Paz), Academic Press, New York 1971, pp. 229-241.

[7] K. C. Y. Mei, "Bridging and stuck-at faults," in Dig. 1973 Int. Symp. Fault-Tolerant Computing, June 1973, pp. 91-94.

[8] A. D. Friedman, "Diagnosis of short faults in combinational circuits," in Dig. 1973 Int. Symp. Fault-Tolerant Computing, June 1973, pp. 95-99.

[9] D. A. Huffman, "Canonical forms for information-lossless finite-state logical machines," IRE Trans. Circuit Theory, vol. CT-6, Special Supplement, pp. 41-59, May 1959.

[10] S. Even, "On information lossless automata of finite order, IEEE Trans. Computers, vol. EC-14, pp. 561-569, Aug. 1965.

[11] Z. Kohavi and P. Lavallee, "Design of sequential machines with fault-detection capabilities," IEEE Trans. Computers, vol. EC-16, pp. 473-484, Aug. 1967.

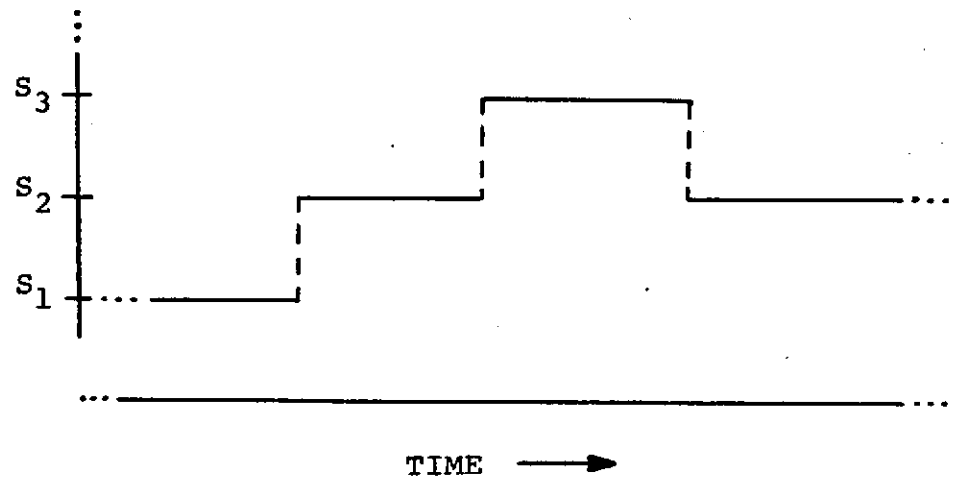


Fig. 1

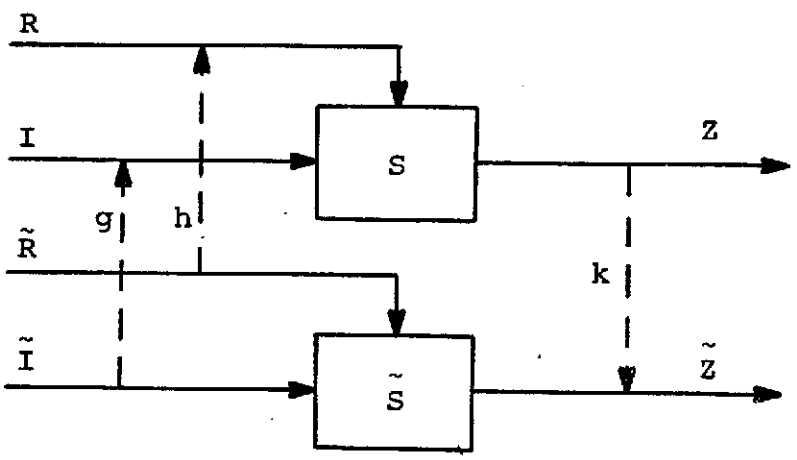


Fig. 2

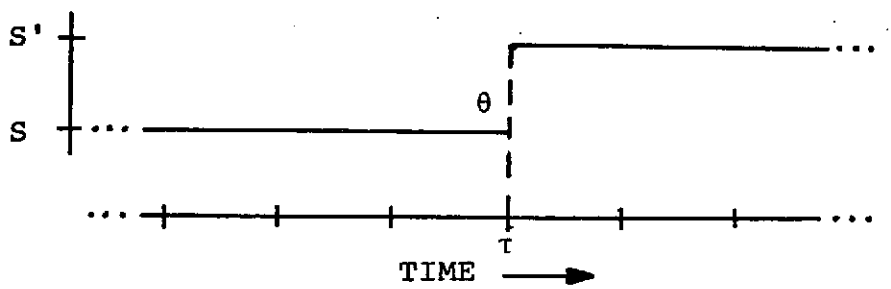


Fig. 3

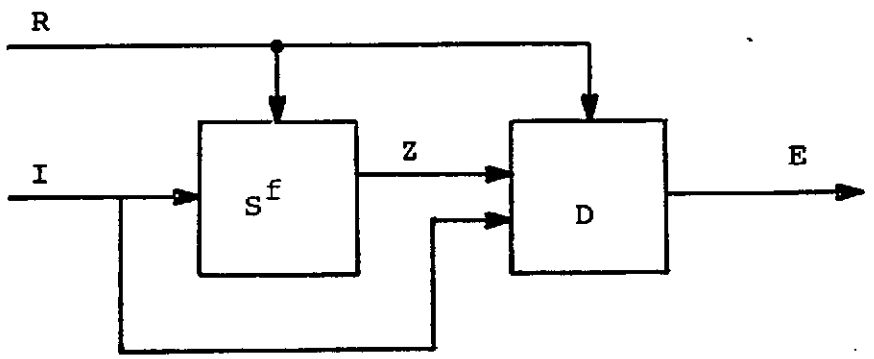


Fig. 4

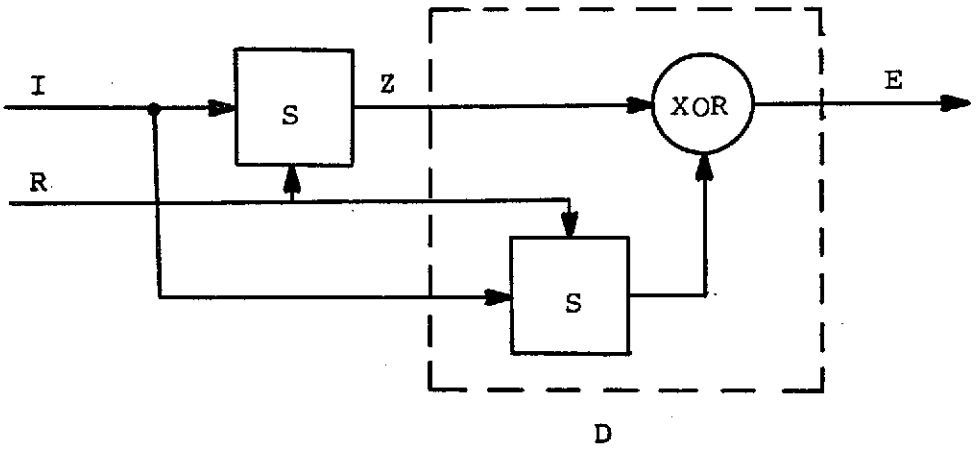


Fig. 5

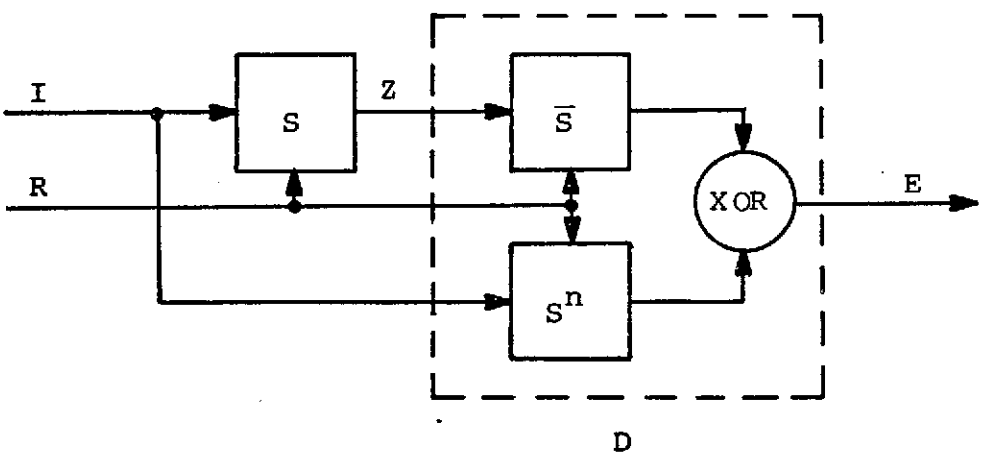


Fig. 6

Q_1 \ I_1	0	1	R_1
a	e/0	f/0	r
b	a/1	b/1	
c	a/0	b/0	
d	e/1	f/1	
e	a/0	c/1	
f	d/1	b/0	

Fig. 7

$\bar{Q}_1$ \ $\bar{I}_1$	0	1	$\bar{R}_1$
A	C/0	D/1	r
B	D/0	C/1	
C	A/0	B/0	
D	C/1	D/1	

Fig. 8

Q_2 \ I_2	0	1	R_2
a	b/0	d/3	r
b	c/1	a/0	
c	d/2	b/1	
d	a/3	c/2	

Fig. 9

$\overline{Q}_2$ \ $\overline{I}_2$	0	1	2	3	$\overline{R}_2$
A	B/0	B/2	B/2	D/1	r
B	A/1	C/0	A/2	A/2	
C	D/2	B/1	D/0	D/2	
D	C/2	C/2	C/1	A/0	

Fig. 10

FIGURE CAPTIONS

Fig. 1. A discrete-time system.

Fig. 2. S realizes $\tilde{S}$ under (g,h,k).

Fig. 3. The result S^f of fault $f = (S', \tau, \theta)$ of S.

Fig. 4. Diagnosis of (S,F) using the detector D.

Fig. 5. Diagnosis via duplication in the detector.

Fig. 6. Diagnosis using an inverse system.

Fig. 7. State table of S_1.

Fig. 8. State table of $\overline{S}_1$.

Fig. 9. State table of S_2.

Fig. 10. State table of $\overline{S}_2$.